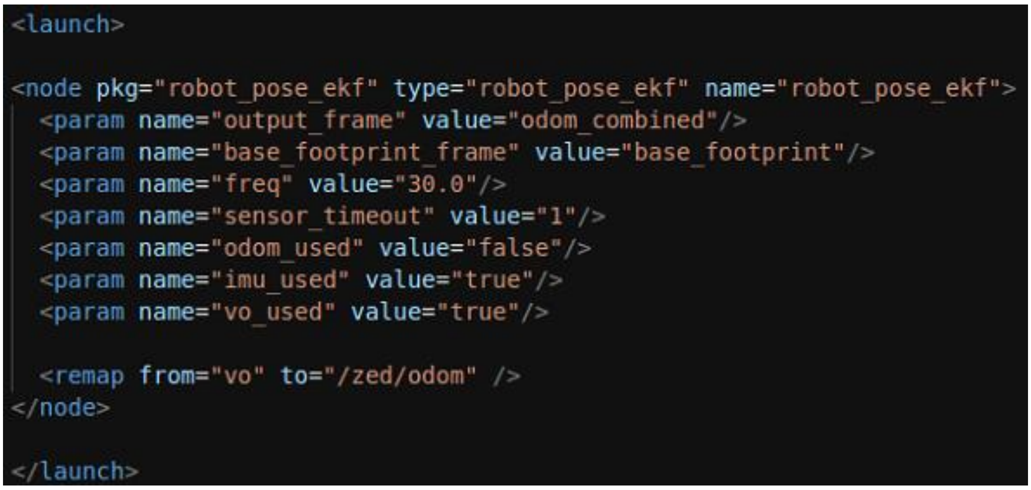
自主駕駛車技術 Self-Driving Cars

林信吉 0510876 2020/04/18

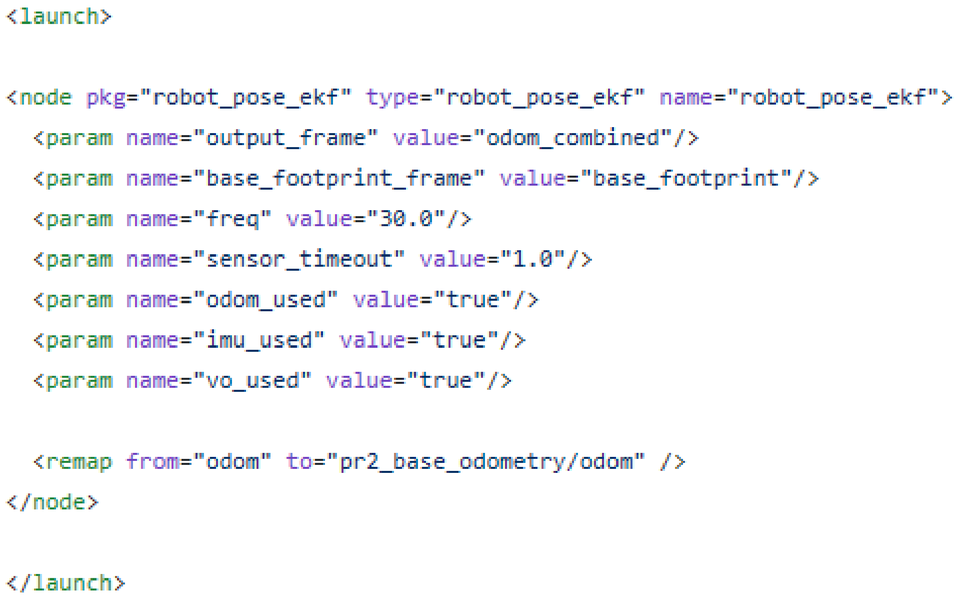
1. What's the difference between our launch (robot_pose_ekf.launch) file and original launch file? And please explain why we add these modification.

➢ robot_pose_ekf.launch (our launch file)



```
<launch>

<node pkg="robot_pose_ekf" type="robot_pose_ekf" name="robot_pose_ekf">
  <param name="output_frame" value="odom_combined"/>
  <param name="base_footprint_frame" value="base_footprint"/>
  <param name="freq" value="30.0"/>
  <param name="sensor_timeout" value="1"/>
  <param name="odom_used" value="false"/>
  <param name="imu_used" value="true"/>
  <param name="vo_used" value="true"/>

  <remap from="vo" to="/zed/odom" />
</node>

</launch>
```

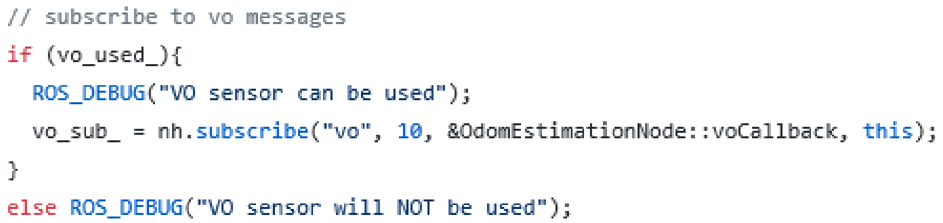
➢ robot_pose_ekf.launch (original launch file)



```
<launch>

<node pkg="robot_pose_ekf" type="robot_pose_ekf" name="robot_pose_ekf">
  <param name="output_frame" value="odom_combined"/>
  <param name="base_footprint_frame" value="base_footprint"/>
  <param name="freq" value="30.0"/>
  <param name="sensor_timeout" value="1.0"/>
  <param name="odom_used" value="true"/>
  <param name="imu_used" value="true"/>
  <param name="vo_used" value="true"/>

  <remap from="odom" to="pr2_base_odometry/odom" />
</node>

</launch>
```

➢ In the robot_pose_ekf launch file, <param name="odom_used" value="true"/> change to <param name="odom_used" value="false"/> is because that the rosbag did not log the wheel odometry data, so we need to change it to false.

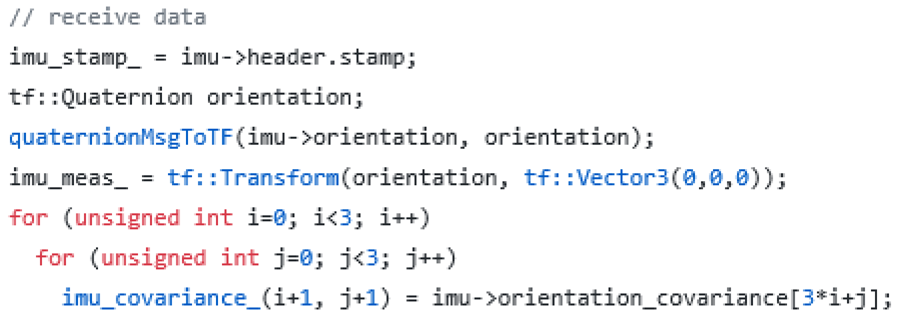
➢ And the second one is <remap from="odom" to="pr2_base_odometry/odom" /> change to <remap from="vo" to="/zed/odom" /> is because we have the visual odometry data from ZED and in the odom_estimation_node.cpp we need to subscribe visual odometry, so we remap topic name from "vo" to "/zed/odom".



```
// subscribe to vo messages
if (vo_used_){
  ROS_DEBUG("VO sensor can be used");
  vo_sub_ = nh.subscribe("vo", 10, &OdomEstimationNode::voCallback, this);
}
else ROS_DEBUG("VO sensor will NOT be used");
```

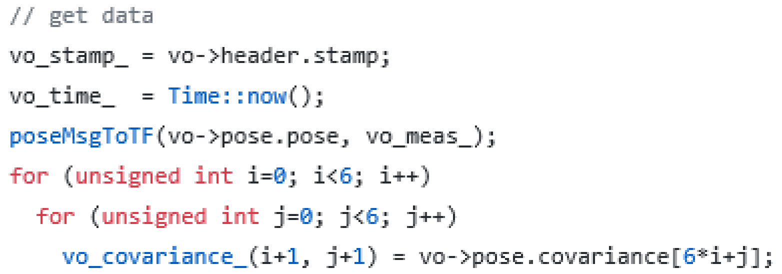
2. **Which parts in IMU data and ZED odometry are used? And please explain why it choose this way. (odom_estimation_node.cpp)**

- IMU data provides orientation information (called function \textbf{tf::quaternionMsgToTF}) while not position information since it is not reliable after double integration from acceleration.



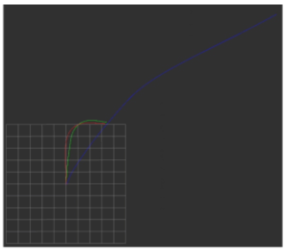
```
// receive data
imu_stamp_ = imu->header.stamp;
tf::Quaternion orientation;
quaternionMsgToTF(imu->orientation, orientation);
imu_meas_ = tf::Transform(orientation, tf::Vector3(0,0,0));
for (unsigned int i=0; i<3; i++)
  for (unsigned int j=0; j<3; j++)
    imu_covariance_(i+1, j+1) = imu->orientation_covariance[3*i+j];
```

- For visual odometry (ZED odometry data) , it provides both position and orientation (called function \textbf{tf::poseMsgToTF}) since it offers relatively reliable position information by visual cues.

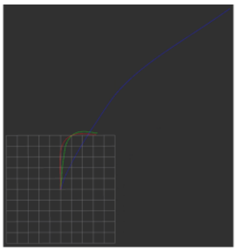


```
// get data
vo_stamp_ = vo->header.stamp;
vo_time_  = Time::now();
poseMsgToTF(vo->pose.pose, vo_meas_);
for (unsigned int i=0; i<6; i++)
  for (unsigned int j=0; j<6; j++)
    vo_covariance_(i+1, j+1) = vo->pose.covariance[6*i+j];
```

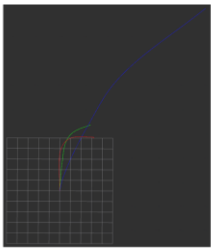
3. **Please try to adjust covariance setting in odom_estimation_node.cpp (in imuCallback() & voCallback()), and observe how it affect the resulting path. Also, give your opinion which setting Is better, and why?**



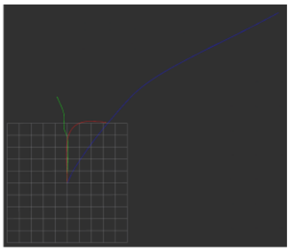
Imu_covariance = 0.0017



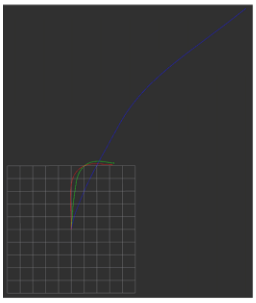
Imu_covariance = 0.017



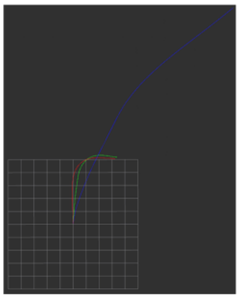
Imu_covariance = 0.17



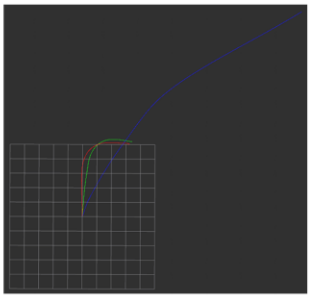
Imu_covariance = 1.7



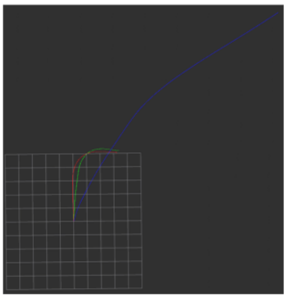
Original
Imu_covariance = 0.00017
Vo_covariance (1, 2, 3) = 0.1
Vo_covariance (4, 5, 6) = 0.17



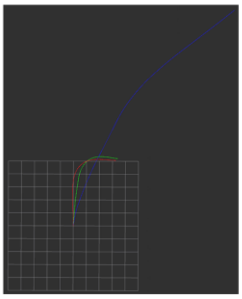
Vo_covariance (4, 5, 6) = 0.017



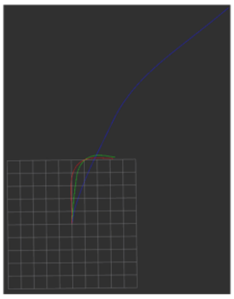
Vo_covariance (4, 5, 6) = 0.0017



Vo_covariance (4, 5, 6) = 0.00017



Vo_covariance (1, 2, 3) = 0.5



Vo_covariance (1, 2, 3) = 0.01

- Since we don't have the ground-truth, we can only qualitatively describe the result. Since only visual odometry provides positional information, no matter how we change the magnitude of position variance, the filtered path will not be affected.
- If we shrink the position variance for Vo, the filtered path will more close to the original one. Similar behavior occurs in the orientation of Vo. In the fourth row, there are no significant differences between the two settings, I guess it is because we only visualize the position but not orientation.
- If we shrink the variance of orientation from Vo, the filtered path will approach to the original one. For IMU, if we enlarge the variance of it, the path will more similar to the original one.
- To answer the problem that which setting is better, we must understand the strengths of each sensor and set up our system correctly (transformation between sensors and robot, etc.). In this dataset, if we make IMU variance even smaller, the path will incline to the left which is incorrect apparently, so I may more believe the data from visual odometry.

4. Comparing the resulting path and the single sensor paths, what is the difference, and why?

- In this assignment, we have two single sensor paths, which are IMU integrated path and Visual odometry path (ZED stereo camera). Combining these two single sensor paths will get the resulting path (combined odometry path). So when we adjust the covariance of IMU or Visual odometry, the resulting path will have the difference output path. This is because we can adjust the covariance setting for IMU and Visual odometry, so first, we must understand the strengths of each sensor and set up our system correctly, after that adjust the covariance setting for IMU or Visual odometry.